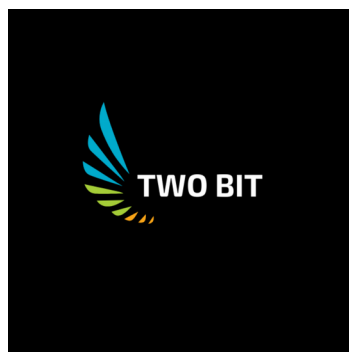
PYTHON PARA INICIANTES

Anderson Jader

PYTHON PARA INICIANTES

Msc. Anderson Jader Antunes Brizola dos Santos

JANEIRO DE 2019

CAP. 1 Introdução ao PYTHON

Tipos de linguagem de Programação

A linguagem de programação define a sintaxe e a semântica que precisamos traduzir nossas ideias em etapas mecânicas. Temos a linguagem de baixo nível e a linguagem de alto nível. Na linguagem de alto nível encontramos as linguagens **compiladas e interpretadas**. O python é um exemplo de linguagem interpretada (Guttag,2013).

Objetos do python

Em um programa ou script nós temos uma sequência de definições e comandos. As definições são avaliadas e os comandos são executados em um interpretador do python que chamamos de Shell. Os comandos e as declarações são quem instruem o interpretador a fazer algo (Guttag,2013).

Tipos de Objetos

Existem dois tipos de objetos, escalar e não escalar.

Escalar: não podem ser subdivididos. (ex.: Números inteiros (int), Números reais (float), vareáveis booleanas (bool))

Não Escalar: possuem uma estrutura interna que pode ser acessada. (ex.: strings).

Guttag, J. **Introduction to computation and programming using Python.** 2013.

Variáveis e Atribuição

Variáveis são aquelas que nos possibilitam criar uma ligação entre um nome e um objeto.

Exemplo:

Queremos encontrar qual é o valor da hipotenusa de um triângulo de catetos b =4 e c = 8

$h = (b^2 + c^2)^{1/2}$

```
b = 4
c = 8
h = ((b**2) + (c**2))**(1/2)
print(h)
```

saída: 8,94427190999916

Mas se mudarmos o valor de C para 20 o que acontece?

```
c = 20
print(h)
```

saída: 8,94427190999916

Obs.: Você pode testar no seu computador.

Interessante voltou o valor inicial de h, por que?

Isso acontece porque quando realizamos a ligação para h ela armazenou o valor de c anterior. Quando vemos o resultado não é uma expressão de valor mas sim um valor particular, então a reatribuição do valor de c não altera o valor da ligação de h.

Strings

As strings são objetos não escalares e são tipadas no shell **‘abc’** ou ainda **str(abc)**, **str(123)**.

Operações com as strings

Somar strings

'a'+ 'b'

saída: 'ab'

Multiplicar

3*'ab'

saída: 'ababab'

Podemos extrair partes das strings

'abc'[0]

saída: ‘a’

Note que as strings são indexadas e o índice inicia a partir do zero.

Podemos realizar *slice* nas strings (a sintaxe é s[inicio:final -1])

'Jader'[2:4]

saída: ‘de’

Podemos calcular o comprimento da string usando a função len()

len('abc')

saída: 3